

Ask me anything

0 questions

0 upvotes

Advanced Design Lab

Today we consider a number of questions concerning good OO design principles. For each question, you will be assigned to a breakout group to discuss your answers. Please post your answers to Mentimeter as usual.

What is Good Design?

good documentation

design by contract

design follows function(-ality)

readability of the code

no duplicated code

efficient code (if we work with big data size)

-A Designe that is easy to understand -
intuitive-Should be easy for a third person to
understand
Designe for the Code

good/initutive naming of methods and
variables

easy to read

What is Good Design?

RDD

easy to test

- good responsibilities- small methods which each only do one thing- Use Assertions- Check for invariants-Code which is easy to test

modular

Well structured class hierarchies

changes and additions can be made easily

avoiding code smells

Group 4:- simple- easy to understand - documentation - similarities to real world

maintainability

What is Good Design?

code coverage with tests

using java documentation

talk to interfaces if necessary

good naming of methods

Restricted visibility for things the user shouldn't be able to access

Polymorphism instead of multiple cases

Task

Print the invoice of Andrei who made a 2 minute national call, a 10 minute international call, 3 local calls at peak time of 2 minutes each, and a local call at off-peak time lasting 20 minutes.

Task

Print the invoice of Andrei who made a 2 minute national call, a 10 minute international call, 3 local calls at peak time of 2 minutes each, and a local call at off-peak time lasting 20 minutes.

Task: Model a system that prints invoices for the customers of a Telco.

- There are 3 types of calls: local, national and international.
- International calls cost 2 CHF / minute (fpm)
- National calls cost 1.50 fpm
- Local calls cost 1 fpm during peak hours, 0.50 fpm off-peak

How would you model the TelCo system? What classes do you propose? How will they collaborate?

we would need an interface for a call the different types of calls would inherit from it, a person class, a invoice object that puts together the calls and calculates costs and the person, a renderer that prints invoice out.

Classes: Company, Customer, Printer, ICall interface, abstract class Call, children of Call with their respective prices (static instance variable) and toString method

Inteface witch gives the functions to a superclass and then subclasses for international, national and local.

Requirements change

- International calls cost 2 fpm peak, 1.5 fpm off-peak
- National calls cost 1.50 fpm peak, 1.25 fpm off-peak
- Local calls cost 1 fpm peak weekend, 0.90 fpm peak weekdays
- Local calls cost 0.50 off-peak

How would you change the TelCo system to meet the new requirements?



Requirements change

- EU calls cost 0.10 fpm
- International (non-EU) calls cost 2 fpm peak, 1.5 fpm off-peak
- National calls cost 1.50 fpm peak, 1.25 fpm off-peak
- Local calls cost 0.90 fpm peak, 0.50 off-peak
- Same-Canton local calls always cost 0.45

What would you change in the TelCo system now?



Last chance for questions